

P2P VIDEO STREAMING WITH INTER-SESSION NETWORK CODING

Jonnahtan Saltarin, Nikolaos Thomos, Eirina Bourtsoulatze, and Pascal Frossard

Signal Processing Laboratory (LTS4)

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

{jonnahtan.saltarin,nikolaos.thomos,eirina.bourtsoulatze,pascal.frossard}@epfl.ch

ABSTRACT

We present a novel receiver-driven p2p system for delivery of multiple concurrent time constrained data streams in overlay networks. We propose an effective combination of rateless coding with intra- and inter-session network coding to efficiently exploit the path diversity in the streaming overlay. Network nodes can decide to forward rateless coded packets or to code them in intra- or inter-session mode before transmission. The transmission strategy is determined based on the availability of data sources and the demands of the children nodes. Each network node solves independently a simple flow maximization problem in order to determine the optimal coding policy. The overall system is evaluated for various networks and the results outline the advantages of the proposed approach over intra-session network coding based schemes in terms of clients' satisfaction, innovative flow rate and decoding delay.

Index Terms— Inter-session network coding, Raptor codes, multicast transmission, overlay networks.

1. INTRODUCTION

The last few years we have witnessed an enormous increase in the demand for novel streaming applications that challenges traditional streaming architectures. Mesh networks and multicast trees are the most common approaches for delivering time constrained data in P2P architectures. Tree-based systems are efficient, however their maintenance is computationally expensive due to network dynamics. Mesh networks are more robust to topology changes and permit the deployment of decentralized maintenance and data distribution algorithms [1, 2]. In mesh networks, the peers exchange messages periodically with their neighbors regarding data availability, *i.e.*, buffer maps. The children peers can then send specific requests to their parents and ask for subsets of available packets in order to maximize the video quality. Such an example is Coolstreaming [2] that is an hybrid pull-push extension where the streams are split into sub-streams and each peer may subscribe to different sub-streams from different neighbors. Network coding [3] can further be used in the overlay nodes in order to benefit from the path diversity in mesh networks. For example, R^2 [1] uses randomized network coding [4] in overlay nodes for transmitting data segments to downstream peers. Such a system is resilient to packet losses and network dynamics and results in a small delivery delay, at the price of frequent buffer map exchanges though.

Coordination of nodes however induces high communication overhead that is often unacceptable when resources are limited. To

alleviate this communication cost LT codes [5] are used in [6] where coordination and packet scheduling becomes unnecessary as only the number of received packets is important in rateless systems. Thus, peer nodes upon receiving a decodable set of packets first recover the data from the received packets and then successively re-encode the data to generate a new set of packets. This however introduces extra delay as nodes should wait to decode the data prior to forwarding it to children nodes. Rateless codes [7] can also be combined with network coding where they permit to avoid requests for specific packets. For example, the need for nodes reconciliation is removed in [8], where nodes perform network coding with rateless encoded packets. Encoding rules are defined in order to efficiently cope with packet losses and bandwidth variations. This system benefits from network diversity and provides robustness to packet losses in a system of peers that all subscribe to the same stream.

In this paper, we propose an effective p2p streaming solution that is able to serve clients with different capabilities. The video stream is encoded with a multiple description coding (MDC) strategy, however the scheme is general and can be used with any other data partitioning strategy, *i.e.*, for video each video frame can be a data segment. The descriptions formed by MDC encoding are then rateless encoded in order to flatten out the relative importance of the different packets in each description and therefore reduce the communication overhead in peer coordination. The multiple descriptions are transmitted concurrently over the network. The quality observed by peers depends on the number of descriptions that they are able to decode. Raptor encoding is applied before transmission in order to increase packet diversity in the network. We propose to implement an inter-session network coding strategy for increased throughput and reduced delivery delays. Inter-session network coding is applied between packets from different descriptions. Peer nodes can therefore choose to transmit rateless coded packets, or to perform intra- or inter-session network coding before transmission. The transmission strategy is defined by the children peers that distributively try to maximize the flow rate based on availability information sent by the parent nodes. We further devise a simple communication protocol where peers indicate data availability to their children and request disjoint sets of packets from their parents. The overall system is tested for various networks and the results outline the benefits of using a combination of Raptor codes and intra-, inter-session network coding strategies. Specifically, the system increases users satisfaction regarding data quality, decreases the delay and enhances packet innovative rate compared to other baseline p2p streaming strategies.

2. INTER-SESSION RAPTOR NETWORK CODING

Network coding [3] leads to increased throughput, decreased delay and decentralized control in networks with diversity. Network nodes

This work has been partly supported by the Swiss National Science Foundation, under grants 200021-118230 and PZ00P2-121906.

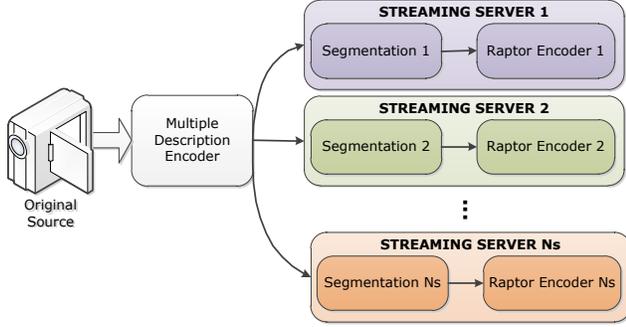


Fig. 1. Data processing at servers. The servers encode the data generated by a MDC encoder. Each segment contains data from a GOP.

typically combine received packets to generate new packets that are transmitted to downstream peers. Clients recover data by means of Gaussian elimination. The network coded packets are augmented with a header of size equal to the number of source (data) packets. This describes the coding operations that each packet has incurred. Many practical issues regarding the application of network coding in real systems have been addressed in [4] where randomized network coding (RNC) is used. Decentralized control is possible as network coding flattens the importance of the packets with only a small performance penalty. Packets are grouped into generations which are packets sharing similar decoding deadlines, *e.g.*, a GOP for video. The generation concept is critical in order to prevent delay explosion and to keep reasonable coding overhead.

Despite the advantages of RNC, the decoding is still performed over a dense equation system as all the source packets are potentially involved in every coded packet. Furthermore, RNC is associated with an overhead that is significant for packets of small size. To alleviate these shortcomings, Raptor network coding has been proposed in [8], where linear network coding concepts are combined with Raptor codes. Specifically, Raptor network codes need a small overhead as the employed Raptor codes have an overhead of two bytes known as Encoding Symbol ID (ESI). This overhead aggregates when packets are combined, however, it remains significantly lower than that of RNC. Furthermore, close to linear decoding times can be achieved by employing 3GPP Raptor codes [9].

In general, network coding is applied between packets from a single source (intra-session network coding). However joint consideration of the multiple sources creates more coding opportunities. This generalization of intra-session network coding is known as inter-session network coding [10]. It finds application in wireless networks as network nodes usually have data from different sources, can overhear others transmissions and interfere with each other.

We propose to explore the design of inter-session network codes for wired overlay networks. We adopt a strategy similar to Raptor network coding approach [8] in order to achieve a low coding overhead and maintain linear decoding and encoding times that are critical for real-time communication. Moreover, we permit combinations between packets from different sources or descriptions of the same video stream. Such combined packets are potentially useful for all the clients interested in one or multiple streams and thus the throughput might be augmented by inter-session network coding. From the perspective of a client that only requests one stream, the combined packets are seen as polluted by the other sources. Therefore, other

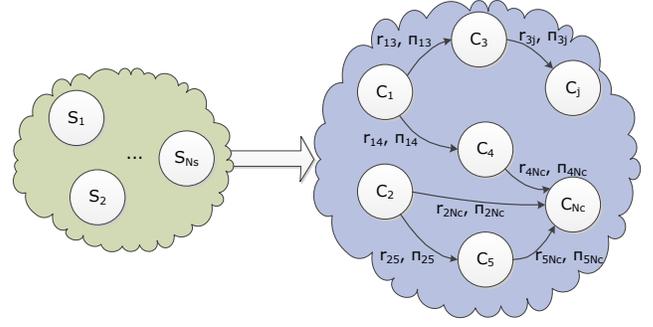


Fig. 2. Mesh networks with N_S sources and N_C clients.

composite packets should be received to clean up the polluted packets.

The selective application of inter-session Raptor network coding results in sparse Raptor codes generator matrices and hence low-cost decoding. Packets are combined only to cope with errors and bandwidth variations. In contrast to RNC inter-session network coding based schemes [10], fewer packets are required for decoding since not all the packets are combined together. When inter-session network coding is applied, the header of the new packet is the concatenation of the composite packets headers, and thus it contains all ESI's and the source identifiers. Similar to [8] the required network coding overhead is small due to both the sparse application of network coding and the small overhead added to the source data packets by Raptor codes. Decoding of inter-session network coding is performed by means of Gaussian elimination. The complete decoding procedure followed at clients is presented in section 3.2.

3. P2P STREAMING SYSTEM

The proposed system consists of multiple servers that stream correlated data. The streams are descriptions of the same time constrained data generated by a video multiple description coder. The descriptions (*i.e.*, sessions) are segmented into generations of K packets. The servers apply non systematic Raptor codes on each generation to augment packet diversity and flatten the relative importance between packets. A header containing the ESI, the generation number, and the description identification number is appended to each packet. We assume that the servers are aware of the end-to-end network conditions and send enough redundant packets to deal with losses and bandwidth variations. Since we target on devising a network coding based data transmission system, we do not deal with the optimal selection of the Raptor codes source rate and we assume that the servers can provide enough redundant packets to cope with losses and bottlenecks. Data processing at servers is depicted in Fig. 1. It should be noted that our scheme is general and can be extended with minor modifications for uncorrelated sources.

3.1. Processing in parent nodes

In the examined network topologies all network nodes act both as clients and forwarding nodes, as illustrated in Fig. 2. The network nodes collect from their parents original Raptor encoded packets or network coded packets. When a node collects a full rank system of

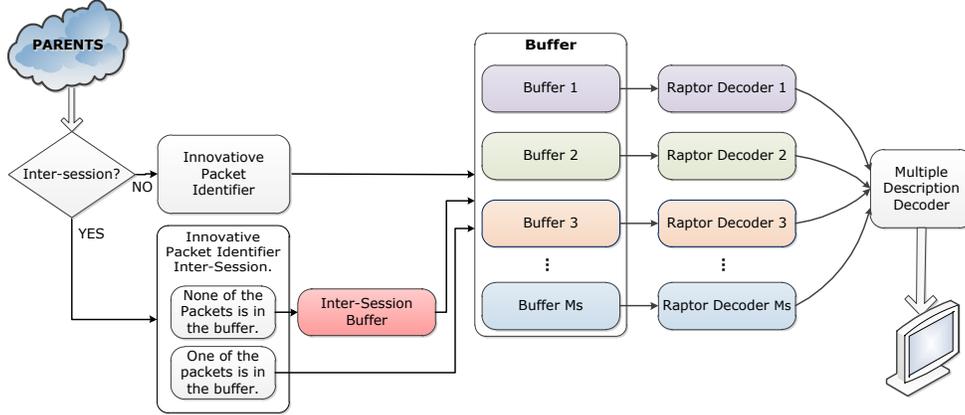


Fig. 3. Innovative packets identification and decoding procedure at clients.

packets, it recovers the data by Gaussian elimination and Raptor encodes the corresponding generation. This maintains a high packet diversity in the network. Finally, at each transmission opportunity, a node has the following options: (a) forward one of the received packets unprocessed, (b) perform network coding within packets from the same description for enhancing robustness to errors, (c) network code packets from different descriptions to improve the throughput, and (d) forward one of the re-encoded packets (Raptor coded packets) generated after Raptor decoding at a node.

3.2. Decoding at clients

Each node decides on the set of packets it requests based on the data availability at its neighborhood and the node’s capabilities, i.e., processing power, display capability etc. The requests are sent to the parent nodes, which forward the requested streams. Each node has as many buffers as the number of descriptions that it receives. In each buffer, only the packets from the corresponding description are stored. Depending on whether a packet is inter- or intra-session network coded different actions are taken. When a packet is intra-session network coded, the packet is examined whether it is innovative with respect to the corresponding buffer. A packet is characterized as innovative only if it brings novel information to the buffer. Only, innovative packets are stored, while non-innovative are rejected.

For inter-session network coded packets a different procedure is followed. First a packet is examined if it is innovative by searching the descriptions’ buffers. When a packet from a single description can be retrieved, the retrieved packet is stored in the appropriate buffer, if it is innovative. Otherwise, the packet is examined regarding its innovation by examining an intermediate buffer called “inter-session buffer”. This buffer contains only inter-session network coded packets that have innovative data for at least one of the descriptions. The packets in the intermediate buffer are re-examined for their innovation (whether all component packets have been recovered or an innovative packet for one of the component buffers has been acquired) each time the description buffers are modified. The overall procedure is summarized in Fig. 3. It should be noted that the packets are flushed from the intermediate buffer when they are redundant for all descriptions’ buffers.

3.3. Communication protocol

The communication between nodes in our system is illustrated in Fig. 4. A node c_j is interested in receiving a set of description D_j , while its parent nodes p_i desire a set of descriptions D_i . The physical link capacity of the link connecting peers p_i and p_j is denoted as r_{ij} . Thus, node c_j should decide how many packets, which ones and from whom it should receive. To this aim, parent nodes p_i advertise to c_j the description availability. Upon receiving this data, node c_j decides about the optimal number of packets from each description to be requested from its parents by solving an optimization problem presented in Section 4. Then, node c_j sends a request message to nodes p_i containing node identity, the requested descriptions $d \in D_j$, the low index $l(i, d)$ and the high index $h(i, d)$ of the requested set of packets of the d th description, where $d \in D_j$. Basically, node c_j tries to maintain high the innovative rate of packets from the set D_j . This enables decoding of the set of descriptions D_j before the decoding deadline imposed by the playback delay.

Nodes start sending packets upon receiving requests from clients. The Raptor coded packets are classified into “common” and “uncommon” set. Streaming servers generate only “common” packets. These packets ESI’s are in the range $[1, N]$ ($N > K$ is the total number of generated packets at servers per generation). Thus, in the beginning of the streaming process only packets from “common” set exist in the network. Packets with larger ESI indices can be generated at network nodes when they collect a decodable set of packets and proceed with Raptor decoding and encoding. Since there is no guarantee that such packets exist in the network, they are called “uncommon”.

To increase the probability that a transmitted packet is innovative for a client, the nodes follow a special policy. When there exists a transmission opportunity a node first searches whether there is a packet from the “uncommon” set that has not been sent. If such a packet is found, it is immediately transmitted. Otherwise, a packet from the “common set” that has been requested but not forwarded previously is sent. If the node cannot find such a packet, the sender transmits packets for error compensation. In this case, an intra-session network coded packet from one of the requested descriptions is formed. Whenever the sender cannot generate a network coded packet (orthogonal packets from the desired descriptions cannot be found), it sends a packet that belongs to the desired descriptions and

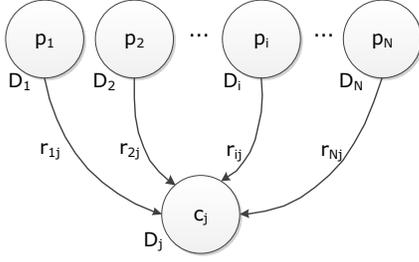


Fig. 4. Peer node c_j desires a set of descriptions D_j , while its parents p_i are interested in possibly other descriptions D_i . The link capacity for a link connecting parent p_i with c_j is equal to r_{ij} .

that has not been sent already.

When there is some available bandwidth for sending other descriptions, the sender decides randomly about the forwarded description and packets from the “uncommon” set are transmitted. If such packets cannot be found, then an inter-session network coded packet is formed and transmitted. Whenever the sender cannot generate an inter-session packet (either the sender has a single description or the sender cannot form a packet that has not been sent), it sends an intra-session network coded packet from the non requested descriptions. Otherwise, it compensates for errors and bandwidth variations following the procedure described above for sending intra-session network coded packets.

4. OPTIMAL PACKET SELECTION

From Section 3, it becomes obvious that nodes should coordinate the requests to their parents in order to collect a decodable set of packets prior to the decoding deadline. We propose here an algorithm that maximizes the requested number of packets from the desired descriptions. Let us denote as n_{id} the number of packets from description d that client c_j requests from parent node p_i . Then, the flow maximization problem is cast as follows

$$\begin{aligned} \max \quad & \sum_{p_i \in P_j, d \in D_j} n_{id} \quad (1) \\ \text{subject to} \quad & \sum_{d \in D_j} n_{id} \leq r_{ij}, \forall p_i \in P_j \\ & \sum_{p_i \in P_j} n_{id} \geq K(1 + \epsilon), \forall d \in D_j \\ & \sum_{p_i \in P_j} n_{id} \geq 0, \forall d \notin D_j \\ & n_{id} \geq 0, \forall p_i \in P_j, d \in D_j \end{aligned}$$

where ϵ is the overhead of the code and K is the number of packets per generation. Since the maximization objective and all constraints are linear, the flow maximization problem is solved by linear programming.

The outcome of the algorithm is a set of n_{id} 's. Therefore, the peer c_j requests the n_{id} packets of the set $[l(i, d), h(i, d)]$ from its

parent p_i , where

$$\begin{aligned} l(i, d) &= \sum_{k=1}^{i-1} n_{kd}, \forall d \in D_j \\ h(i, d) &= l(i, d) + n_{id}, \forall d \in D_j \end{aligned}$$

where $l(i, d)$ and $h(i, d)$ are the lower and the upper indices of the packet set requested from the node p_i .

5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed scheme, we consider the transmission of multiple description coded data consisting of three descriptions. The descriptions are balanced, *i.e.*, every description offers the same quality. Each server forwards a single description. In total, each server sends 100 Raptor encoded packets per generation from which 20 are redundant packets to cope with packet losses and network bottlenecks. These packets form the common set. The optimal selection of the number of the redundant packets is an interesting problem however it is out of the scope of this paper. The packet size is set to 1000 bytes. We compare the performance of the proposed inter-session network coding scheme (InterNC) with that of the Raptor network coding scheme presented in [8] (RaptorNC) in terms of:

- **Dissatisfaction:** The difference between the number of descriptions a peer requests and the number of descriptions that it recovers. It is averaged over all generations and normalized with respect to the total number of requested descriptions.
- **Delay:** Time duration needed for a peer to decode a generation. When a peer cannot meet the decoding deadline (1000 ms), the delay is set to infinite (2000 ms).
- **Innovative Flow:** The number of received innovative packets from the requested descriptions per generation normalized over the delay.

For the sake of comparison two types of networks are considered: (a) clustered networks and (b) networks from PlanetLab project [11]. The clustered networks consist of two sub-networks connected to each other with a few links like in Fig. 5. Each sub-network is an irregular network generated from a regular by removing some links and redirecting some others as suggested in [8]. Essentially, the upper sub-network serves as data provider to the lower sub-network. The servers are connected to the second sub-network with low capacity links (remedy links) that cannot provide sufficient bandwidth to the clients in the lower sub-network. These links provide Raptor coded packets to decode inter-session network coded packets. All the simulations are performed in NS3 [12]. For all the networks, we randomly select the number of descriptions (and which ones) each client aims at receiving.

5.1. Clustered networks

We consider three streaming servers and all other nodes to be clients. The upper sub-network consists of 9 nodes while the lower of 12 nodes. The irregular networks are the result of removing 10% of the links and changing the destination for 15% of the links from regular networks. The regular networks have three nodes per coding stage and each node in a coding stage is connected with all nodes from the previous coding stage. The link capacity in each sub-network is 2 Mbps, while the links connecting the servers with the upper sub-network have capacity 1 Mbps. Finally, the links connecting the

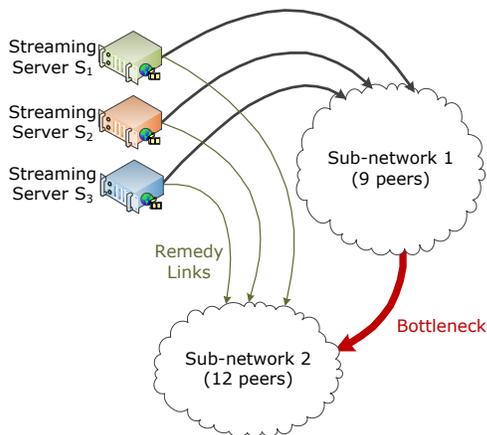


Fig. 5. Clustered network with remedy links.

two sub-networks have capacity 1 Mbps and the remedy links have capacity 500 kbps.

We first investigate the performance of both schemes under comparison in terms of dissatisfaction. The results with respect to the packet loss rate are presented in Fig. 6 (a). All the links suffer from the same loss rate which takes values in the range $[0, 20]\%$. We can see that the average dissatisfaction for the whole network is lower for InterNC than that of RaptorNC scheme. The performance difference increases with the loss rate as with InterNC packet combinations from different descriptions can be formed, while RaptorNC often sends redundant packets when non orthogonal packet combinations cannot be found. Interestingly, from Fig. 6 (a) we note that the advanced performance of InterNC is driven by the performance over the lower sub-network, while for the upper sub-network both schemes perform equally well. This is attributed to the remedy links that provide intra-session network coded packets to help decoding of inter-session network coded packets. Apparently, the bottlenecked links forward more inter-session network coded packets.

We reach similar conclusions by examining the performance of InterNC and RaptorNC for the examined clustered networks regarding their innovative flow rate. The results are illustrated in Fig. 6 (b). We can observe that in overall InterNC performs slightly better than the RaptorNC scheme and that higher innovative rate is achieved for InterNC at the lower sub-network, while for the upper sub-network the situation is reversed and RaptorNC outperforms InterNC. The upper sub-network is penalized by the application of InterNC as there are not enough original Raptor coded packets to decode the combined packets. Such packets are provided to the lower sub-network by the remedy links.

5.2. PlanetLab networks

The results over clustered networks have shown that the proposed InterNC scheme is interesting for streaming data. Here, we evaluate our scheme for more realistic network topologies captured from PlanetLab project [11]. The network topology consists of three servers that stream different descriptions and 27 clients. The capacities of the links vary from 27 kbps up to 10871 kbps. We place randomly 6 nodes that are interested in decoding one description, 13 nodes that want two descriptions and 8 that request all three descriptions. All the links experience the same loss rate.

As previously, we test the performance of both schemes in terms of dissatisfaction. Fig. 7 (a) depicts the performance of nodes requesting one, two, or three descriptions. We observe that InterNC and RaptorNC perform similar for nodes demanding a single description, while significant gains can be noticed for all other nodes. This is attributed to the fact that inter-session network coded packets cannot assist when a node wants to decode one description since it does not receive enough packets to decode the combined packets. For nodes that want to decode two descriptions, inter-session network coded packets become useful; however for higher error rates these nodes cannot receive enough remedy packets. Finally, nodes that require all descriptions can find more packets to decode inter-coded packets and thus are more robust to higher loss rates.

For the sake of completeness, we examine also the performance of InterNC and RaptorNC regarding the decoding delay. From Fig. 7 (b), we can note that the results are in agreement with the conclusions reached for the dissatisfaction performance. Therefore, gains for InterNC exist for nodes demanding two or three descriptions, while the advantages of InterNC are diminished when a single description is requested. Interestingly, the proposed InterNC allows decoding from a smaller system of packets in comparison to other RNC based inter-session network coding schemes where the number of collected packets should be at least equal to the cumulative number of source packets of the component sources.

6. CONCLUSIONS

We have presented a novel p2p streaming system employing intersession network coding based on rateless codes. The proposed scheme is appropriate when multiple concurrent streams flow over the network. The scheme is receiver driven and requires minimal communication overhead. The network nodes determine the optimal coding policy by solving a flow maximization problem that optimizes the bandwidth usage. The overall system is evaluated for various networks and the results highlight the advantages of the proposed approach over intra-session network coding based schemes in terms of innovative flow rate, dissatisfaction measure and delay.

7. REFERENCES

- [1] M. Wang and B. Li, " R^2 : Random Rush with Random Network Coding in Live Peer-to-Peer Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.
- [2] S. Xie, B. Li, G. Y. Keung, and X. Zhang, "Coolstreaming: Design, Theory, and Practice," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1661–1671, Dec. 2007.
- [3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [4] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. 41st Allerton Conf. on Communication Control and Computing*, Monticell, IL, USA, Oct. 2003.
- [5] M. Luby, "LT codes," in *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, Vancouver, Canada, Nov. 2002, pp. 271–280.
- [6] C. Wu and B. Li, "rStream: Resilient and Optimal Peer-to-Peer Streaming with Rateless Codes," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 77–92, Jan. 2008.

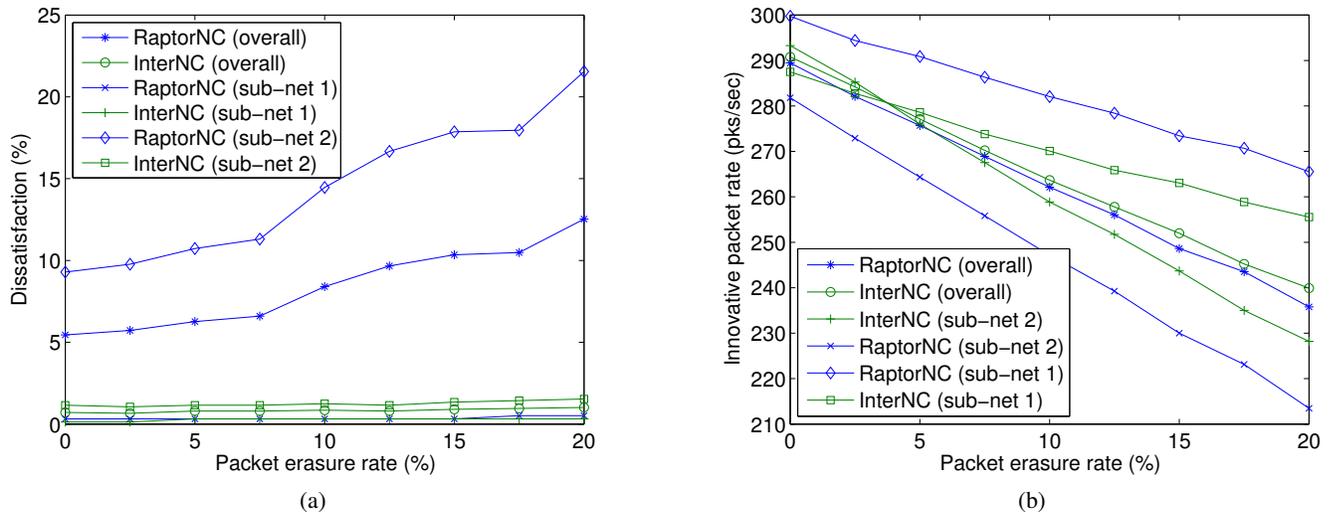


Fig. 6. Performance comparison of InterNC and RaptorNC for the examined clustered networks with 24 nodes regarding their (a) dissatisfaction and (b) innovative flow rate.

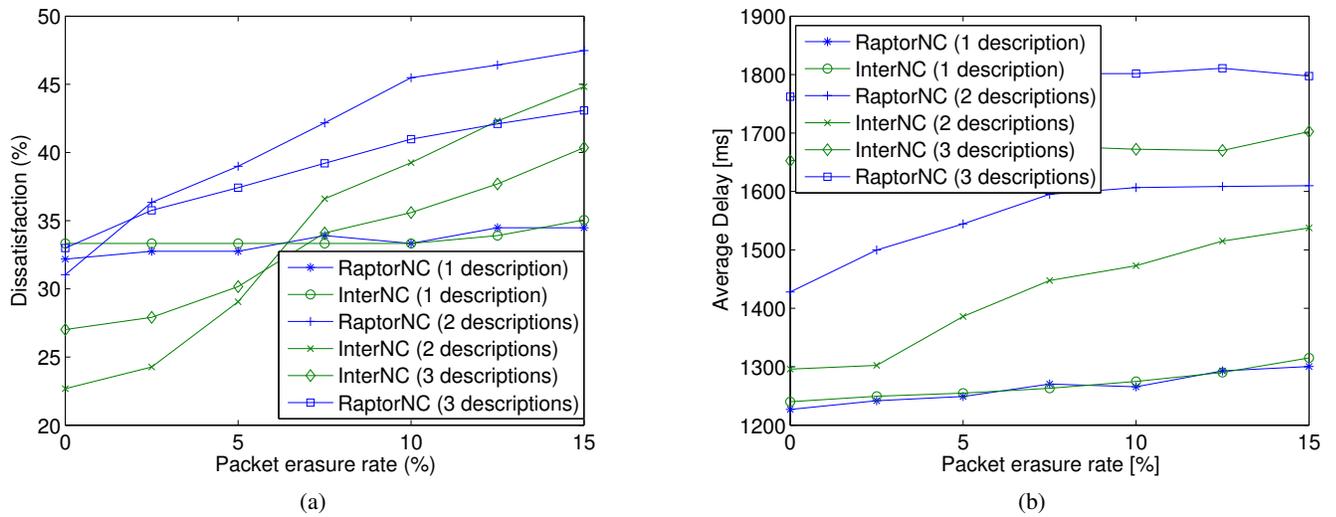


Fig. 7. Performance comparison of InterNC and RaptorNC for transmission over PlanetLab networks with 3 servers and 27 clients: (a) nodes dissatisfaction and (b) delay. The results are the averages of nodes requesting one, two, or three descriptions.

[7] A. Shokrollahi, "Raptor codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[8] N. Thomos and P. Frossard, "Network Coding of Rateless Video in Streaming overlays," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1834–1847, Dec. 2010.

[9] "3GPP TS 26.346 V7.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs," June 2005.

[10] Y. Wu, "On Constructive Multi-Source Network Coding," in *IEEE Int. Symp. on Information Theory*, Seattle, WA, USA, July 2008, pp. 1349–1353.

[11] "Planetlab:," [Online], Available on web site <http://www.planet-lab.org/>.

[12] "The network simulator - ns3," [Online], Available on web site <http://www.nsnam.org>.